

**Universidade Federal de Sergipe
Departamento de Computação**

Engenharia de Software II

Profª Drª Adicinéia A. de Oliveira

Avaliação Arquitetural – um roteiro

2025-2



Existe um formulário ATAM?

- **ATAM (Architecture Trade-off Analysis Method)** é um dos métodos mais clássicos e práticos para avaliação arquitetural.
- Existe um formulário padrão?
 - **Sim e não.**
- O ATAM não tem um *formulário único oficial* (como um template fechado), mas ele define **estruturas de coleta de informação** e passos bem definidos, com artefatos esperados em cada fase.
- Existem modelos, planilhas e checklists frequentemente utilizados por praticantes, mas cada organização pode adaptar.



Existe um formulário ATAM?

- Os artefatos principais que podem ser organizados em formulários são:

Artefato

Cenários (Quality Attribute Scenarios)

Lista de Stakeholders

Atributos de Qualidade (Quality Attributes)

Lista de Decisões Arquiteturais

Matriz de Sensibilidade e Trade-off

Conteúdo

Formulário com: Estímulo, Fonte de Estímulo, Artefato, Ambiente, Resposta e Medida de Resposta

Nome, papel, interesse, influência

Ex.: Desempenho, Segurança, Usabilidade, Manutenibilidade, Disponibilidade etc.

Decisão, justificativa, trade-offs esperados

Relação entre decisões e atributos de qualidade



Template Simplificado do ATAM

a) Identificação dos Stakeholders

Nome	Papel	Interesse	Influência
Ex.: João	Usuário	Disponibilidade	Alta



Template Simplificado do ATAM

b) Lista de Atributos de Qualidade

Atributo de Qualidade	Descrição
Desempenho	Tempo de resposta baixo
Segurança	Proteção contra ataques
Disponibilidade	Tempo de atividade
Manutenibilidade	Facilidade de modificação



Template Simplificado do ATAM

c) Cenários de Qualidade (usar para cada atributo)

Atributo	Fonte de Estímulo	Estímulo	Ambiente	Artefato	Resposta	Medida de Resposta
Desempenho	Cliente	1000 acessos simultâneos	Horário de pico	Servidor Web	Tempo de resposta mantido	< 2s



Template Simplificado do ATAM

d) Decisões Arquiteturais Identificadas

Decisão

Uso de cache

Load balancing

Justificativa

Reduzir latência

Suportar mais conexões



Template Simplificado do ATAM

e) Matriz de Sensibilidade e Trade-off

Decisão	Desempenho	Segurança	Disponibilidade
Cache	+	0	+
Load Balancer	+	-	+

Legenda: + (contribui), - (impacta negativamente), 0 (neutro)

Trade-off > Troca compensatória — ou seja, ao escolher uma opção, você necessariamente abre mão (total ou parcialmente) de outra.



Template Simplificado do ATAM

e) Matriz de Sensibilidade e Trade-off

Decisão	Desempenho	Segurança	Disponibilidade
Cache	+	0	+
Load Balancer	+	-	+

Legenda: + (contribui), - (impacta negativamente), 0 (neutro)

A decisão de usar **load balancing** melhora **desempenho** e **disponibilidade**, mas pode dificultar a **segurança** (porque aumenta a superfície de ataque e a complexidade de gestão de sessões, por exemplo).

Esse é o **trade-off**: você **ganha em duas dimensões**, mas **perde em uma**.



Exemplo de aplicação do ATAM

- Sistema:

- Aplicativo de agendamento de consultas online para um hospital universitário.

Objetivos:

- Permitir agendamento 24h/dia.
- Reduzir filas de espera.
- Garantir privacidade dos dados.

Atributos Críticos:

- Disponibilidade.
- Segurança.
- Desempenho.
- Escalabilidade.



Exemplo de aplicação do ATAM

Exemplo de Cenário:

- Atributo: Disponibilidade.
- Fonte de Estímulo: Paciente.
- Estímulo: Tenta agendar às 2h da manhã.
- Ambiente: Servidor em manutenção.
- Artefato: Sistema de agendamento.
- Resposta: Apresenta mensagem amigável de indisponibilidade planejada.
- Medida de Resposta: Manutenções programadas notificadas com 24h de antecedência.



Exemplo de aplicação do ATAM

Possíveis Decisões Arquiteturais:

- Adotar infraestrutura em nuvem com redundância.
- Política de backup automático.
- Criptografia ponta-a-ponta para dados sensíveis.



Matriz de Débito Técnico

Matriz de Débito Técnico (*Technical Debt Quadrant*), proposta por Martin Fowler.

Eixo Horizontal (Intenção):

Prudente: A equipe está ciente e faz escolhas conscientes, mesmo sabendo que está gerando um débito técnico.

Imprudente: A equipe toma decisões sem planejamento ou consciência das consequências.

Eixo Vertical (Conhecimento):

Deliberado: A decisão foi tomada intencionalmente.

Inadvertido: A decisão foi consequência de falta de conhecimento ou entendimento.



Matriz de Débito Técnico



Matriz de Débito Técnico - exemplo

Deliberado

Prudente

“Precisamos lançar agora e lidar com as consequências”Exemplo: O cliente exige a entrega de um MVP (mínimo produto viável) em 2 semanas. A equipe decide usar uma solução simples, sabendo que terá que refatorar o código depois para escalabilidade e segurança.

“Agora sabemos o que deveria ter sido feito”Exemplo: Após alguns meses de produção, descobre-se que o framework de persistência não suporta bem o volume de dados atual. Na época da decisão, a equipe não tinha essa informação.

Imprudente

“Nós não temos tempo para design”Exemplo: A gestão força a entrega rápida sem considerar qualquer impacto técnico. A equipe implementa várias soluções improvisadas sem planejamento, acumulando bugs e riscos graves.

“O que estamos fazendo?”Exemplo: A equipe não entende a tecnologia, usa más práticas, copia código da internet sem revisão, acumulando problemas sérios de manutenção e segurança sem sequer ter consciência disso.

Inadvertido



Exemplo de Matriz de Débito Técnico

Matriz de Débito Técnico (Baseada no Mini Caso):

Tipo	Prudente	Imprudente
Deliberado	<p>"Vamos usar nuvem e backup automático agora e ajustar a escalabilidade depois". <i>Exemplo: A equipe opta por uma arquitetura mínima viável na nuvem, sabendo que precisará escalar futuramente.</i></p>	<p>"Não temos tempo para configurar segurança avançada, depois vemos isso". <i>Exemplo: Decisão apressada ignorando práticas de segurança.</i></p>
Inadvertido	<p>"Agora sabemos que deveríamos ter considerado balanceamento de carga desde o início". <i>Exemplo: Só após problemas de desempenho a equipe percebe essa necessidade.</i></p>	<p>"Por que o sistema caiu?" <i>Exemplo: Falta de monitoramento e conhecimento da arquitetura levou à queda sem explicação.</i></p>



Atividade em Grupo

- Cada grupo considerando o projeto prático que já estão desenvolvendo em Engenharia de Software II, deve:
 - a) Identificar stakeholders.
 - b) Listar 3 atributos de qualidade.
 - c) Descrever 1 cenário para cada atributo.
 - d) Listar 2 decisões arquiteturais.
 - e) Elaborar a matriz de trade-off.
- Cada grupo apresenta em 5 minutos suas análises.



Sugestão de planilha

- **Aba 1:** Stakeholders (com as colunas do item a).
- **Aba 2:** Atributos de Qualidade (item b).
- **Aba 3:** Cenários (item c).
- **Aba 4:** Decisões Arquiteturais (item d).
- **Aba 5:** Matriz de Trade-off (item e).



Referências

PRESSMAN, Roger - Engenharia de Software, Uma Abordagem Profissional. 8ª Edição. McGraw Hill Education, 2016.

SOMMERVILLE, Ian. Engenharia de Software. 9ª Edição. São Paulo: Addison-Wesley, 2007.

https://edisciplinas.usp.br/pluginfile.php/4641855/mod_resource/content/1/Aula%209%20-%20Padr%C3%B5es%20Arquiteturais.pdf

https://edisciplinas.usp.br/pluginfile.php/4647423/mod_resource/content/1/Aula%2010%20-%20Avalia%C3%A7%C3%A3o%20Arquitetural.pdf

<https://www.devmedia.com.br/como-identificar-divida-tecnica-em-projetos-utilizando-mineracao-de-dados/32497>

<https://productoversee.com/conhecendo-o-debito-tecnico-no-desenvolvimento-de-produtos/>





Obrigada!!!

Profa. Dra. Adicinéia A. Oliveira
adicineia@dcomp.ufs.br

 ProfaAdicineiaOliveira

Perguntas?

